UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

GRADO EN SISTEMAS DE INFORMACIÓN

Trabajo Fin de Grado

Diseño e Implementación de una Herramienta Visual que Obtenga el Modelo Relacional Lógico a partir de un Diagrama Entidad Relación

Autor: Gonzalo San Miguel Oteo

Tutor/es: Iván González Diego

TRIBUNAL:

Presidente:

Vocal 1º:

Vocal 2º

FECHA:

INDICE

1. Resumen

Actualmente en el mundo de la empresa, o a nivel usuario se sabe qué es una base de datos, para qué sirve y que servicios puede llegar a darte. No hay ninguna empresa que pueda realizar tareas de gestión de almacenamiento de la información a mayor o menor escala que no utilice una base de datos, es un pilar fundamental para llevar un orden y control. Pero no todo el mundo conoce como se construye una base de datos ordenada y coherente, que pasos se siguen, como se tiene que estructurar y por qué se estructura de esa manera. Cuando se construye una base de datos lo primero que se tiene que saber es que se va a almacenar, como se va a distribuir en las tablas, como o con qué atributo o clave primaria se van a relacionar… toda esta información se obtiene realizando un diccionario de datos.
La aplicación que se va a realizar será capaz de:

- Realizar un diagrama de Entidad-Relación, un pequeño diagrama donde se dibujarán las entidades con sus atributos, sus claves primarias y relaciones, para tener un primer esquema de la Base de Datos.

- Y convertir automáticamente el diagrama Entidad-Relación en un modelo Relacional-Lógico, este modelo se caracteriza por mostrar nuestra base de datos en formato tablas, con nuestros atributos y relaciones.

2. Summary

Currently, in the business world or at user level it knows what a database is, for what it is used and what services it can provide. There is not any company performing information storage management tasks in greater or less scale that is not already using a database.
It is key to perform with rigor and control. However not everyone knows how a tidy and consistent database it is built, what steps are followed, how it must be structured and why it is structured in that way. When a database is going to be built, the first thing to know is what is going to be stored, how the tables are going to be organized and how or what attribute or primary key is going to be linked... You can get all this information designing a dictionary of data.
The application that is going to be implemented will be capable to:

- To make a diagram of entity-relationship: a small diagram to show the entities with their attributes, the primary keys and relations, in order to have a first scheme of the Data Base.

- And to convert automatically the diagram entity-relationship into a Relational-Logic model. This model is characterized because is able to show our database in tables format, with their attributes and relationship.

3. Palabras Clave

- Entidad/Relación: Tipo de diagrama de Base de Datos
- BB.DD: Bases de datos
- Java: Lenguaje de Programación
- Modelo Lógico Relacional: Tipo de diagrama de Base de Datos

4. Glosario de Acrónimos y Abreviaturas

- E/R: Entidad Relación
- BB.DD.: Base de Datos
- PK: Primary Key
- PFK: Primary Foreing Key
- FK: Foreing Key

5. Resumen Extendido

Durante este trabajo se ha estudiado a fondo el proceso de Diseño de las BB.DD. Como se ha dicho dentro del mundo laboral es uno de los pilares básicos para llevar una buena organización y mantenimiento de la información, tener una buena gestión y diseño de la base de datos.

Para conseguir una buena funcionalidad y diseño de la base de datos se necesita:

- Un diccionario de datos → No suele ser habitual este paso en las empresas, ya que los campos que se van incluyendo en las tablas suelen ir creciendo poco a poco o modificando, se establecen unos campos predefinidos por seguridad para llevar un registro de quién hace un insert, delete, update y las PK de cada tabla para no haber repeticiones de registros.

- Diseñar un diagrama Entidad Relación → es uno de los diagramas más importantes ya que se puede visualizar a primera vista, como será y como ira cogiendo forma nuestro esquema final de nuestra BB.DD. En este diseño se dibujarán las entidades con forma de rectángulos, atributos con forma de circunferencias que irán unidas a las entidades y por último las relaciones serán rombos, que indicarán el tipo de relación.

Figura .1 Ejemplo de diagrama E/R

- Diseñar modelo Lógico Relacional → sería la conversión del modelo entidad/relación. Con este modelo se pueden ver las tablas con los campos, PK, FK, PFK, relaciones, tipos de relaciones, y si tienen relaciones (n,m), se verá la tabla resultante… sería el diagrama final y el más detallado de la base de datos. Con el siguiente ejemplo se puede ver una solución final, del modelo lógico relacional. Hay numerosas formas de diseñar este modelo y diferentes terminologías.

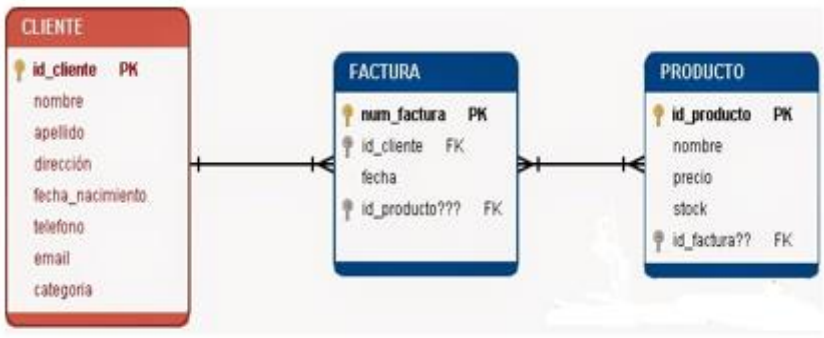

Figura 2. Ejemplo de modelo L/R

Una vez hecha una breve introducción de estos modelos, paso a hacer un pequeño resumen de mi proyecto.

Se ha creado una aplicación en lenguaje Java con la plataforma NetBeans, se ha decidido utilizar este lenguaje por la versatilidad que da a la hora de construir programas y las ayudas y funcionalidades. Teniendo varias librerías que podrían ayudarme a la hora de construir figuras geométricas, líneas, rectángulos, rombos, circunferencias… También se pensó en este lenguaje, ya que es uno de los que más he profundizado a lo largo de la carrera y además como aprendizaje personal. La aplicación consiste en construir desde cero un modelo Entidad-Relación y automáticamente convertirlo al diagrama Lógico-Relacional.

Se ha intentado usar ideas de diseño y funcionalidad de aplicaciones similares, pero estudiando el mercado se ve que no hay ninguna aplicación similar que te haga este tipo de conversiones.

Si se pueden ver aplicaciones donde se puede crear un modelo Entidad-Relación o un diagrama Lógico-Relacional, pero ninguna que desde un modelo Entidad-Relación se obtenga el Lógico-Relacional automáticamente.

Cuando se estudia este tipo de programas se ve que el usuario tiene que realizar todo el diseño de los modelos o el diagrama, un ejemplo fue el programa DIA, donde hasta el usuario tiene que dibujar la raya y la circunferencia de los atributos e ir arrastrándolo hasta la entidad, esto me dio

la idea de automatizar ese proceso: crear una pantalla donde poder asignar los atributos y los datos de la entidad y al darle aceptar se auto dibujara todo en la entidad.

Con las relaciones entre entidades la forma más eficaz fue parecida, cuando se tienen las entidades dibujadas y con todos sus datos introducidos, marcar con el ratón la entidad que se quiere unir y a cuál, el proceso de diseñar el tipo de relación y sus posibles atributos dependiendo de la relación se realiza igual que las entidades: se introducen los datos necesarios, aceptar y saldría todo dibujado.

En la conversión se utiliza el mismo método, dependiendo de lo introducido por el usuario se construye automáticamente el diagrama resultante Lógico-Relacional. Es más, a medida que se va construyendo el modelo Entidad-Relación se puede ir viendo cómo va quedando el diagrama Lógico-Relacional, ya que permite ir del modelo al diagrama y viceversa.

Los pasos que se han seguido y que más adelante se irá contando y detallando, para la creación de la aplicación una vez elegido el lenguaje de programación, son los siguientes:

- Estudio del diagrama entidad-relación.
- Estudio del modelo lógico-relacional.
- Estudio del tipo de relaciones y conversiones PK, FK, PFK.
- Estudio de aplicaciones similares.
- Diseño visual de la aplicación.
- Diseño de la funcionalidad de la aplicación.
- Diseño del código de la app.
- Depuración del código.
- Pruebas de la aplicación.

Una de las fases más costosas fue el diseño del código, en concreto la parte de cómo hacer las figuras geométricas, cómo realizar las relaciones, las líneas de entidad a entidad y justo que se corten en el punto medio del lado más cercano a la entidad que se quiere unir, hacer los rombos en medio de las relaciones, que se dibujen los atributos en los bordes de la entidad o del rombo de la relación.

Es uno de los métodos más importantes donde se utilizan ecuaciones matemáticas, la posición constante del ratón, y de las entidades en el panel de diseño. Para poder realizar estas características, más adelante se explica en profundidad éste método, con partes de código utilizado.

Se utiliza también los formularios de Java para realizar el diseño del panel, ésta es una de las utilidades que da Java y facilita la construcción visual de la aplicación.

A lo largo de esta memoria se verá con más detalle y con fragmentos de código cómo se ha ido creando poco a poco la aplicación. Ha sido un proyecto agradable y que he cogido por gusto ya que se aprende bastante del lenguaje de java y de bases de datos.

6. Memoria:

6.1. Objetivo:

El objetivo principal del trabajo consiste en llevar a cabo el estudio, diseño, planificación y puesta en marcha de una aplicación en JAVA donde se podrá dibujar un diagrama de E/R y al pulsar un botón se generará automáticamente el modelo relacional lógico.

Objetivos específicos planteados para lograr resultados satisfactorios son los siguientes:

1. Decidir el lenguaje con el que se va a hacer la aplicación.
2. Recopilar información sobre el lenguaje que se ha elegido.
3. Desarrollar un primer boceto/diseño de la aplicación.
4. Desarrollar el código del panel donde se va a dibujar nuestro diagrama de E/R.
5. Realizar las pertinentes pruebas de nuestro panel.
6. Estudiar las diferentes normas de conversión del diagrama de E/R al Modelo Relacional Lógico.
7. Desarrollar el código de conversión al Modelo Relacional Lógico.
8. Realizar las pertinentes pruebas de conversión.
9. Evaluar los resultados obtenidos y poner en marcha nuestra aplicación.
10. Redactar un informe o memoria final.

Antes de comenzar a crear la aplicación, se necesitó un estudio previo de lo que se quería construir, conocer que es una base de datos, para qué sirve y qué funcionalidad se le puede llegar a dar.

Conocer bien la funcionalidad de las tablas y las características de sus atributos, centrándose en tres tipos de atributos que pueden tener y son importantes a la hora de construir la aplicación, al igual que conocer las propias características de cada diagrama a dibujar.

6.2. Diagrama Entidad Relación:

Es el diagrama que se va a dibujar y el que se va a convertir una vez terminado. A continuación, se va a describir y a comentar cada una de las partes de las cuales está compuesto.

Diagrama Entidad Relación → Denominado con sus siglas E/R, es una herramienta para el modelo de datos que permite representar las entidades relevantes de un sistema de información. Está compuesto por:

- Entidad → Son el elemento fundamental de este modelo, las entidades son las llamadas tablas de una base de datos, cuando se realiza la conversión. Las entidades pueden representar cosas en concreto como una persona, un coche… o abstractas como un préstamo, una reserva…, las entidades se dibujan con rectángulos.

- Atributos → Se representan mediante un círculo. Un atributo es identificativo y recoge una característica o propiedad de la entidad y si es un atributo clave o primary suelen ser dibujados con un círculo negro o subrayado. A continuación, se describen los tipos de atributos.

    - PK (Primary Key) → Se llama PK a un atributo o la combinación de varios atributos para la identificación única de una entidad.

- Relaciones → Se representan con un rombo etiquetado en su interior con un verbo. Este rombo se une mediante una línea a dos entidades, estas relaciones pueden ser de diferente tipo y tienen diferentes cardinalidades.

    - Tipos de Relaciones:

        - Relación Identificativa → Las relaciones Identificativas, cuando una entidad no puede formar su PK con sus propios atributos, produce que la PK de otra entidad esté formada por la PK de la otra relación más alguno de sus propios atributos. En la conversión, pasa la PK a la entidad relacionada como PFK, formando parte de la tabla como un campo más que no es propio.

        - Relación No Identificativa →Las relaciones no identificativas sólo asocian o relacionan dos entidades. Cuando se produce la transformación, la PK

pasa a la tabla relacionada como FK, formando parte de la tabla como un campo más. Únicamente se produce en el modelo relacional y sólo asocia o relacionan dos entidades.

- Tipos de cardinalidad:

  - (1,1) → Relación uno a uno (ejemplo → Una persona tiene un DNI).

  - (1,n) → Relación uno a muchos (ejemplo→Una persona tiene varias casas).

  - (n,m)→ Relación de muchos a muchos (ejemplo→ Un cliente puede tener varios productos, y un producto lo pueden tener varias personas), este tipo de cardinalidad permite a las relaciones poder tener atributos, creando una tabla auxiliar en la conversión de este modelo. Esta tabla auxiliar sus PK serán las PK de las tablas unidas por ésta relación, y se convertirán como PFK en el modelo relacional.

6.3. Modelo Lógico Relacional:

Este modelo es el resultante al convertir el diagrama E/R. El modelo de datos relacional organiza y representa los datos en forma de tablas o relaciones. Este modelo no se distingue entre tipos de entidades y tipos de relaciones ya que la idea es que una relación o tabla expresa la relación entre los tipos de valores que contiene.

Este modelo tiene diferentes conceptos al E/R:

- Tablas →Es el factor resultante de la conversión de las entidades más los atributos.
- Atributo → Se les llama a los campos o columnas de la relación. Su nombre debe de ser simple, no pueden ser nombres compuestos. Dependiendo del tipo de relación:
    - PFK (Primary Foreing Key) o FK (Foreing Key) → Los atributos PFK o FK, dependiendo del tipo de relación, entre dos o más tablas. Son aquellos atributos PK que se heredan de una a otra tabla y que son necesarios para hacer referencia a varias filas de la tabla a la que se hace referencia. Formando finalmente en la tabla hija la PK más la PFK o FK.
- Esquema de una relación → Se compone por el nombre de la relación y una lista de atributos.
- Conjunto de entidades → Relación o tabla.
- Claves → Son las PK de las entidades en el E/R
- Instancia de una relación → Conjunto de entidades, cada entidad se representa como una tupla. Cada componente de la tupla corresponde al valor del atributo correspondiente, según el orden enunciado en el esquema de la relación.

6.4. Resumen esquema:

Entidad / Relación	Lógico Relacional	
Entidad.	Tabla.	
Atributo.	Columna / Campo.	
Identificador Único.	Clave Primaria.	
Relaciones N:M	Nueva tabla con clave primaria la concatenación de las claves primarias de las entidades que la forman.	Esta diferenciación se debe a que todas las claves ajenas deben hacer referencia a las claves primaria de otras tablas y consecuentemente no pueden ser nulas. Dicho de otra manera, toda referencia ajena debe hacerse a un campo único. Es la integridad referencial.
Relaciones 1:M	Propagando la PK de 1 en la de muchos (creando un campo en la de muchos que referencie a la de 1). Si cada elemento de la entidad que participa con muchos aparece en la entidad de uno, es decir, si TODOS los elementos de la entidad de muchos tienen asociado uno de la entidad de uno.	
Relaciones 1:1	Propagar la PK (igual que en la de 1:M). Si cada elemento de cada entidad que participa se asocia con sólo uno de los elementos de la otra entidad. Da igual donde se propague.	

Tabla I. Proceso de conversión o equivalencia entre el modelo E/R al modelo Relacional.

6.5. Resultados, Metodología y Recursos Utilizados.

6.5.1. Resultados obtenidos:

El principal resultado del trabajo será una memoria y una aplicación en Java con los siguientes contenidos:

- Análisis detallado del funcionamiento de la aplicación.
- Análisis detallado de partes del código importante de la aplicación.
- Aplicación en Java para la conversión de diagramas E/R a Modelo lógico Relacional.
- Un manual completo para el usuario.

6.5.2. Metodología utilizada:

- Se hace un primer diseño de la aplicación que se va realizar, para concretar qué lenguaje es el más adecuado para la construcción: en nuestro caso será Java.
- Se realiza un primer estudio de las clases de Java más útiles, para la construcción de la aplicación, se investiga sobre el modelo E/R y el diagrama Lógico Relacional, conversiones, diseños…
- Se hacen los primeros bocetos de la aplicación, para que sea visible, automática e intuitivamente, para facilitar al usuario en todo momento.
- Se diseñan las clases, métodos, relaciones de nuestro código.
- Y una vez diseñado y estudiado las normas de los cambios de un diagrama a otro, se comenzará a crear la aplicación.
- Cuando se tiene construida se realizan las pruebas pertinentes para obtener los posibles fallos, correcciones o mejoras.

6.5.3. Recursos Utilizados:

Para la realización del proyecto se precisará disponer toda la información necesaria sobre las Bases de Datos, procesos de conversión de Entidad Relación a Lógico Relacional y librerías Java para la creación de la aplicación.

También se requerirá el siguiente equipamiento informático:

- Hardware: Se necesitará un pc personal, acceso a Internet, y un procesador de alta velocidad para realizar la búsqueda de toda la información, a la vez del desarrollo de la aplicación en Netbeans.
- Software: Se necesitará Netbeans, programa de diseño para la creación de figuras geométricas o algún otro tipo de diseño visual en la aplicación. Word para la realización de la memoria final.

6.6. Análisis

A continuación, se va analizar la aplicación que se ha desarrollado.

6.6.1. Requisitos

- Conocer las normas de conversión, para poder realizar la conversión de forma automática.
- Crear un panel donde poder ir dibujando nuestro modelo E/R.
- Crear dos ventanas de propiedades, una para las entidades y otra para las relaciones.
- Aplicación intuitiva, automatizada y fácil de usar para un usuario novel en este entorno.

6.6.2. Análisis de Requisitos

Cuando se han definido los requisitos esenciales que tendrá la aplicación se puede comenzar con el desarrollo y creación para cumplirlos.

- Normas de conversión: Este punto es de los más importantes en la aplicación, por un lado, es fácil ya que tras recoger toda la información que el usuario ha insertado en su proyecto, únicamente se tendría que ir transformando cada dato para el modelo resultante. Por ello en necesario conocer las normas de conversión:

    - Regla 1: Por cada tipo de entidad fuerte o débil del modelo E/R se crea una relación que contenga todos los atributos simples y no multivaluados de la entidad. Como clave primaria (PK) de la relación se formará con los atributos que formen parte de la clave primaria de la entidad.

    - Regla 2: Por cada tipo de relación del modelo E/R de cardinalidad 1:1, se elige una de las entidades donde se incluirá la PK de la otra entidad.

- Regla 3: Por cada tipo de relación del modelo E/R de cardinalidad 1:N se identifica la entidad que representa el lado N de la relación y se incluye como clave foránea (FK), (PFK) si es relación de tipo identificativa, la clave primaria (PK) de la entidad que pertenece al lado 1 de la relación.
- Regla 4: Por cada tipo de relación de modelo E/R de cardinalidad N:M se creará una nueva tabla, la cual contendrá las claves primarias (PK) de las entidades relacionadas, formando la PK de la nueva tabla.

- Crear un panel y ventanas de Propiedades: El diseño principal de la aplicación fue que todo estuviera en una misma pantalla: panel de dibujo, botones de opciones... por lo que se ha tenido que optimizar el espacio para poder tener un panel de dibujo amplio para el usuario.

  Al crear las dos ventanas de propiedades se da al usuario de ver la información no destacada y se reduce la información en el panel ayudando a una mejor visualización, a la hora de convertir el E/R ocurrirá lo mismo, se muestra la información más relevante.

- Intuitiva, automatizada: Al querer crear una aplicación donde el usuario tenga que hacer lo menos posible a la hora de dibujar, únicamente se pensó poner un único botón para añadir entidades. No se pensó, ni se veía factible que el usuario tuviera que dibujar la simbología de los atributos en las entidades, la recta de la relación, o redimensionar entidades, ya que se quería buscar la facilidad siempre desde cualquier punto para el usuario. Por ello el programa recoge toda la información que se introduce y cada vez que se pulsa Aceptar se hace un repaint () del proyecto.

6.7. Diagrama de Flujo

Se va a detallar el diagrama de flujo de la aplicación, estudiando los diferentes caminos que se puede tener hasta llegar a la solución.

Leyenda

Figura 3. Figuras para representar el fuljo del programa

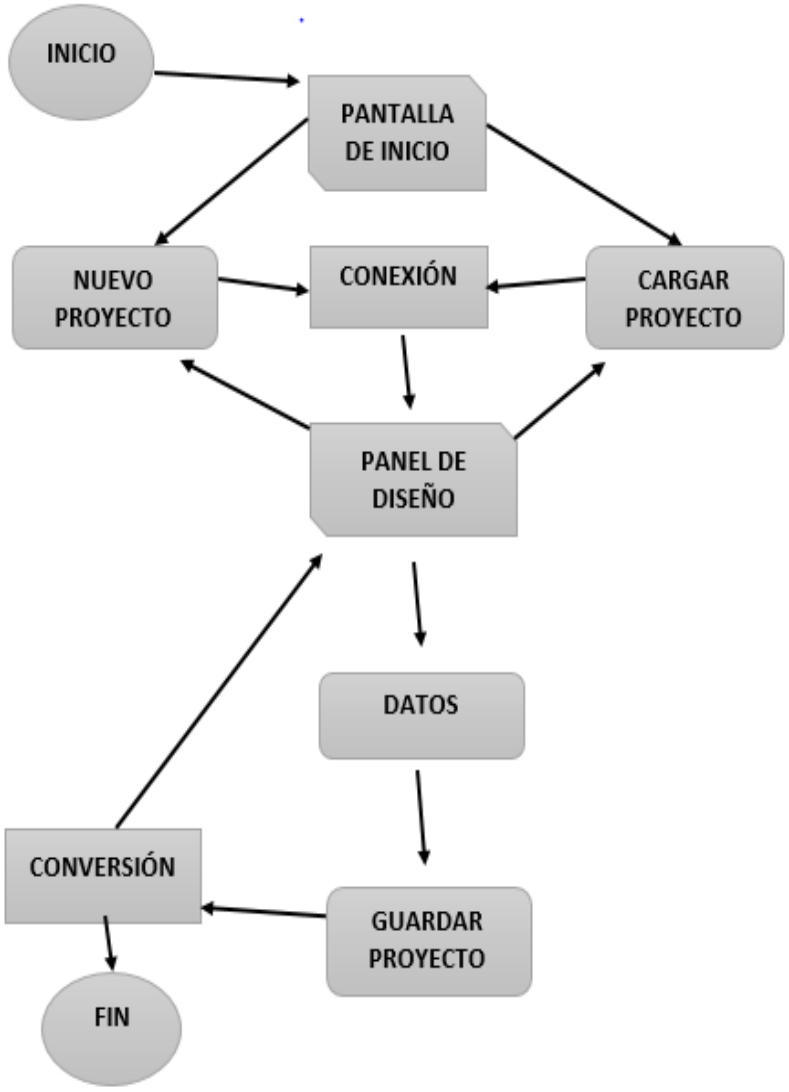

Figura 4. Diagrama de flujo

A continuación, se comenta paso a paso el flujo.

INICIO ➔ Se lanza la aplicación.

PANTALLA DE INICIO ➔ Primera pantalla que se verá en la aplicación, desde ella se tienen dos opciones, crear un nuevo proyecto o cargar un proyecto ya existente.

CONEXIÓN ➔ Una vez elegida una de las opciones, la aplicación generará el proyecto elegido o creará uno nuevo dirigiendo al usuario al panel de diseño.

PANEL DE DISEÑO ➔ En esta pantalla se verá nuestro proyecto cargado o nuestro nuevo proyecto, también se podrá cargar uno nuevo, crear uno nuevo e introducir nuestros datos para el modelo E/R.

DATOS ➔ Son los datos necesarios para crear el modelo E/R una vez terminado, se podrá guardar el proyecto.

GUARDAR PROYECTO ➔ Una vez terminado el proyecto se guarda, y se podrá convertir.

CONVERSIÓN ➔ Se genera la conversión del proyecto que esté en el panel de diseño. Como se ha explicado se podrá volver al modelo E/R pudiendo modificar el proyecto, guardarlo o crear uno nuevo.

FIN ➔ Se finaliza la aplicación.

6.8. Arquitectura e Implementación

A continuación, se va a explicar la arquitectura de la aplicación, nombrar y comentar las clases más importantes.

Se comenzará mostrando un UML de la aplicación, pero debido a su complejidad se va a ir mostrando por trozos para explicarlo mejor.

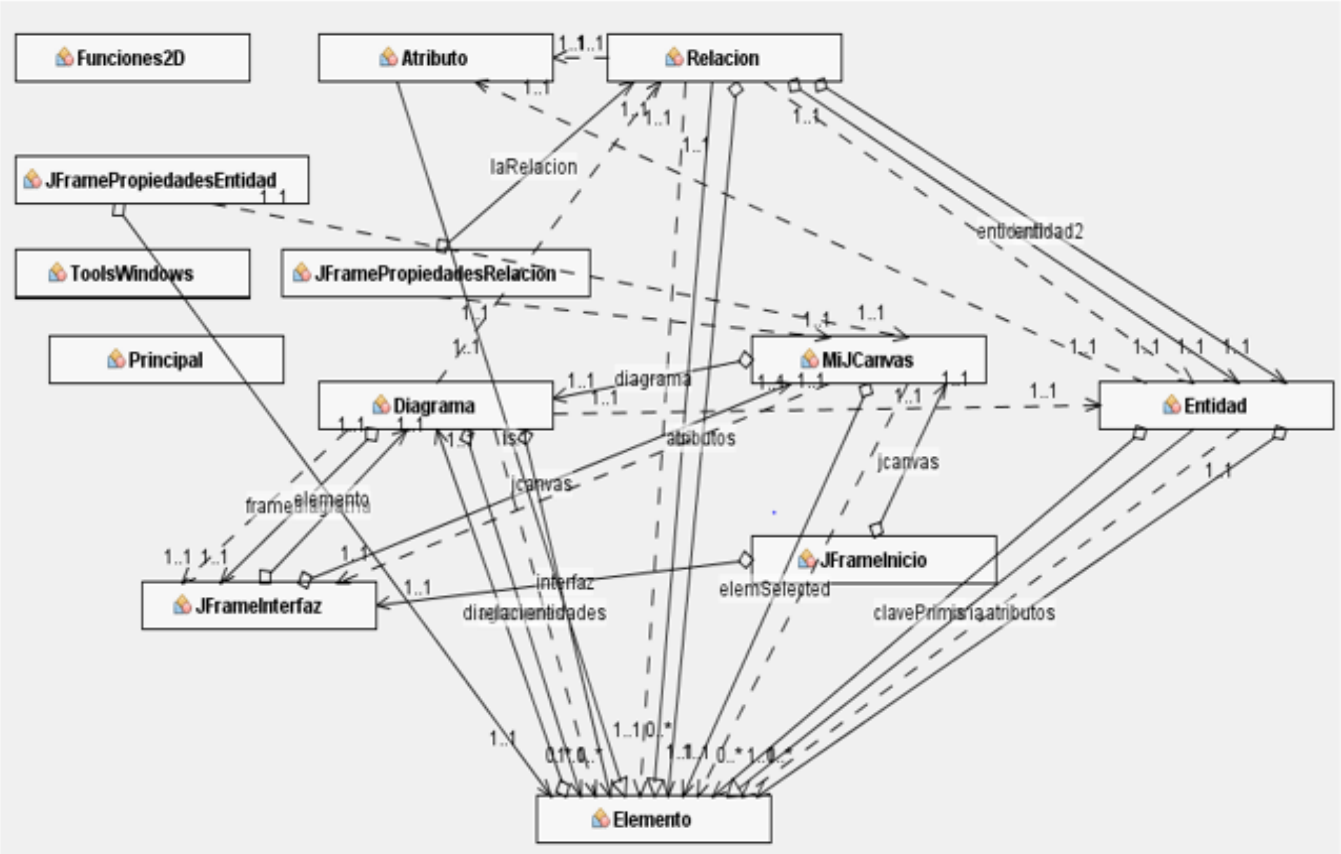

Figura 5. UML

La aplicación está dividida en cuatro paquetes: Auxiliares, Elementos, Interfaz y Recursos.

6.8.1. Auxiliar

Figura 6. Paquete Auxiliar

El paquete Auxiliar está compuesto por:

Funciones2D

Comenzaré comentando una de las clases más importantes a la hora de dibujar en el panel de forma automática, las relaciones, entidades, rombo de las entidades… En ella se han desarrollado los métodos necesarios, para la creación de los polígonos de forma correcta, puntos de corte, puntos medios… a continuación se explicará cada uno de los métodos. Para esta clase se debe saber que se utilizará siempre las coordenadas de la posición de las entidades, que se han enumerado los vértices de la entidad de cero a tres en sentido horario y la actual y final del ratón.

- PuntosIncidenciaRectaPoligono → En este método se calcula el punto de incidencia de la recta de la relación con el polígono de la entidad, cuando se crea una relación se localizan los puntos de coordenadas de la entidad inicial y de la entidad final, sabiendo el punto exacto de corte.

- DistanciaSegmento → para calcular la longitud de la recta, sabiendo los puntos de corte en cada lado de la entidad.

- GetEcuancionGeneralRectaPasa2Puntos → Uno de los métodos más laborioso ya que se han utilizado fórmulas matemáticas, en este caso la Ecuación general de la recta dada por dos puntos, donde Ax = (y2-y1)x; By = (x1-x2)y; C = x1*(y1-y2) + y1*(x2-x1) y se obtiene A, B y C que se usará en el siguiente método.

- GetEcuacionGeneralRectaPerpendicularPorPunto → Una vez obtenidos los valores anteriores se puede saber la perpendicular a esa recta gracias a otra fórmula matemática, la pendiente de la recta. Con ello se sabe el punto medio de la recta y así se puede dibujar más adelante el rombo utilizando ese punto.

- GetPoligonoByPunto → Una vez obtenido ese punto medio sumando una distancia variable, desde el centro a sus 4 puntos cardinales (-x, -y), (x, y), (-x, y) y (-y, x) se obtiene el eje de un rombo y únicamente se tiene que dibujar los lados sabiendo ya los puntos finales.
- getPoligonoFlecha → En este método, se realizarán los mismos pasos que para la obtención del rombo, pero esta vez solo se dibujarán dos lados en forma de flecha.

- SegmentoSaleAtributo → Con este método y sabiendo el lado de la entidad donde siempre se quiere que salga el segmento y el círculo de la clave primaria, únicamente se debe pasar el punto medio y dibujar una recta vertical hacia abajo una distancia variable.

Dispone de más métodos auxiliares a estos para la ayuda de sumas, multiplicaciones o divisiones.

ToolsWindows

Esta clase se ha creado para ajustar el frame a la resolución de la pantalla donde se ejecute la aplicación.

6.8.2. Elementos

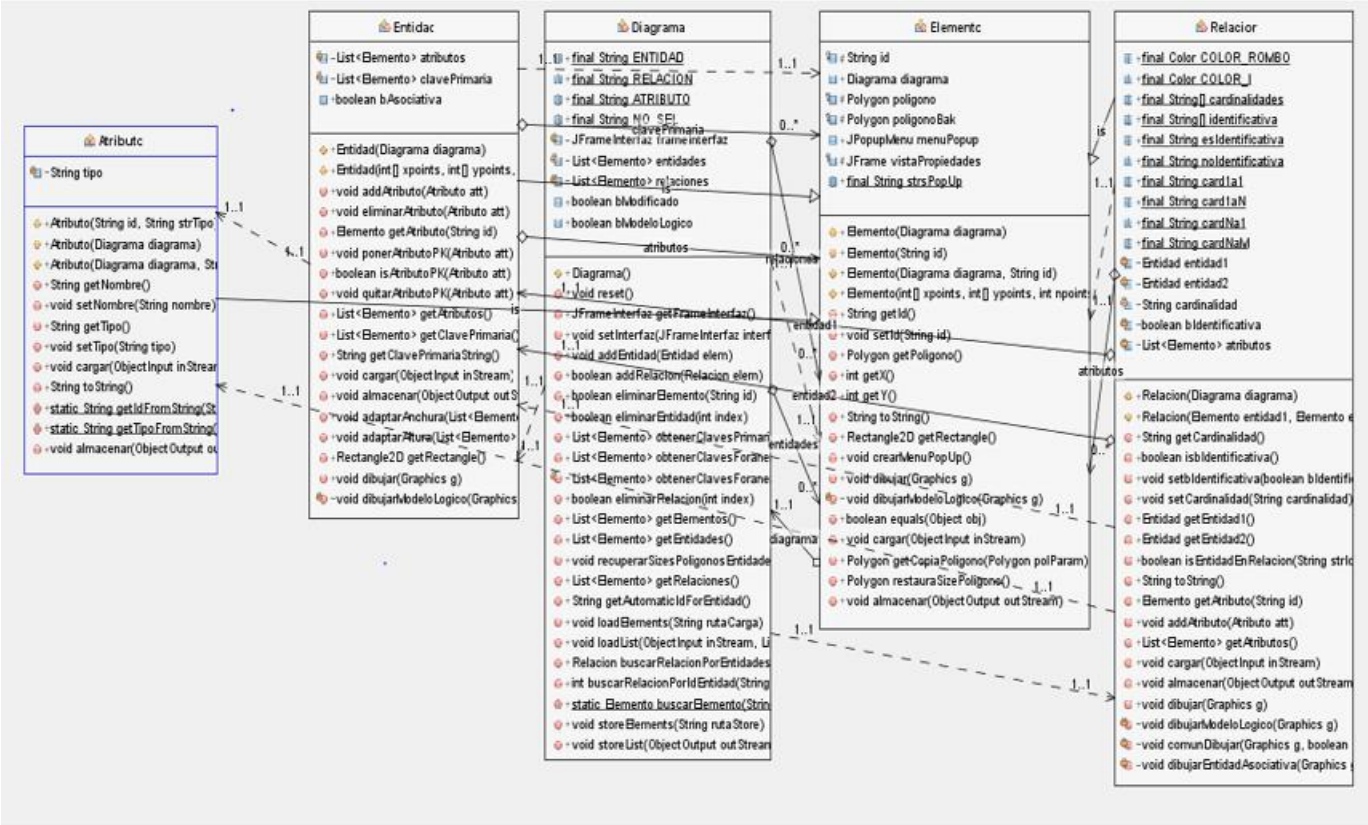

Figura 7. Paquete Elementos

El paquete Elementos está compuesto por:

Diagrama

Esta clase está compuesta por varias ArrysList:

- ObtenerClavesPrimariasForaneas → Donde se almacenan las claves Foráneas para a la hora de la conversión poder reordenarlas en la entidad correctamente, con la simbología correcta.

- ObtenerClavesForaneas → Donde se almacenan las claves primarias sin realizar el cambio de la simbología.

- ObtenerClavesForaneasRelaciónNM → Lista única donde se almacenan las claves primarias unificadas de las entidades correspondientes en una relación N:M.

- GetElementos, getEntidades, getRelaciones → Donde se almacenarán las entidades dibujadas en el panel, los elementos utilizados (polígonos, rectas, rombos…) y por último las relaciones utilizadas.

Otros métodos importantes que están disponibles en esta clase serían los siguientes:

- AddRelacion → Se comprobará que se ha iniciado una unión entre dos entidades, y si es correcta se creará una nueva relación.

- EliminarEntidad → En este método se comprueba la entidad marcada, si tiene una o más relaciones y si tuviera alguna se borrarán a la vez que la entidad, esta búsqueda se hace mediante un ID que se autogenera cuando se crea una nueva entidad.

- EliminarRelación → Este método es más sencillo que EliminarEntidad, ya que únicamente se borra la relación sin depender de eliminar una entidad, es decir no se borra una entidad si se borra la relación.

- LoadElements y loadList → Estos dos métodos serán los encargados de cargar los datos (entidades, relaciones, atributos…) y los elementos (rectas, polígonos) de un fichero dándole una ruta. Una vez cargados se mostrarán en el panel de diseño.

Atributo

En ésta clase se tendrán los métodos relacionados con los atributos y sus propiedades, es la clase más sencilla y simple:

- Cargar → Leer el atributo del fichero.

- Almacenar →Se almacena el atributo en el fichero sin el tipo de atributo que se reconstruirá cada vez que se cargue un fichero.

- ToString → Se fusiona el Atributo con el Tipo

Entidad

En esta clase se tendrán los métodos relacionados con las entidades y sus atributos:

- AddAtributo, EliminarAtributo → Con estos dos métodos se añadirán y eliminarán los atributos de la pantalla de propiedades de la entidad. Se crean excepciones por si el atributo está repetido, o si no se señala nada a la hora de eliminar.

- PonerAtributoPK, QuitarAtributoPK → Con estos dos métodos al hacer doble click en la lista de los atributos se añadirá a la lista de claves primarias o se eliminará.

- ArrayList: Atributos, ClavePrimaria → En estas dos listas se guardarán todos los atributos de la entidad incluyendo la PK de la entidad, y en la otra lista únicamente las PK.

- Almacenar, Cargar → Estos dos métodos, almacenan y cargan los atributos correspondientes de cada entidad, contienen un bucle FOR para ir recorriendo todos los atributos posibles, antes se inicializa con un contador.

A continuación, se comentarán los métodos necesarios para el pintado de las entidades y de sus atributos:

- AdaptarAnchura, AdaptarAltura → Estos dos métodos redimensionarán la entidad para que se adapte a la hora de la conversión, donde se pintarán todos los atributos y claves dentro de la tabla resultante. Para ello se cuentan los caracteres del atributo más largo y se multiplica por un tamaño establecido dando así un tamaño nuevo a la entidad.

- Rectangle2D → Este método dará el tamaño inicial a la entidad en la posición establecida con el mouse.

- Dibujar →Método con el que se dibujará todo el cuerpo de la entidad, junto con su PK, las características establecidas, color, relleno… Se utiliza una librería llamada Graphics. Antes de realizar el proceso se controlará con un IF si el modelo a dibujar el Lógico o el E/R. ya que si es el Lógico se irá al método que se comentará a continuación. Este IF depende del botón de conversión: si está marcado o no, así permitirá volver de modelo a otro. Recordar que únicamente en el E/R se muestra el nombre de la tabla, y su clave primaria.

- DibujarModeloLogico →En este método se pintarán todos los atributos dentro de la entidad con su tipo, se eliminará el segmento y la circunferencia que indican la clave primaria en el E/R, y seguirá aplicando el mismo color de la entidad, pero diferente tamaño ya que llamará a los métodos que redimensionan la entidad. También en éste método se pintarán las PFK y las FK de las entidades relacionadas si las hubiera. Para ello se utilizarán dos bucles FOR, y las listas creadas, anteriormente comentadas.

Relación

Esta clase es parecida a la clase Entidad a la hora del pintado. Sus métodos son parecidos, pero también tiene métodos propios:

- DibujarEntidadAdociativa → En este método se configura la creación de una entidad virtual que aparecerá como tabla a la hora de la conversión, utilizando la librería de Graphics2D y Rectangle2D. Una vez se tienen los puntos de donde se posiciona el centro del rombo, se calcula la línea de unión y se dibuja la tabla.

- ComunDibujar → En este método se encuentra el dibujado común al E/R y al Lógico Relacional. Se controla también el tipo de cardinalidad de la relación, para el dibujado de la flecha. Ya que ésta apunta siempre a la entidad (N)

- Almacenar,Cargar → Mismos métodos que en la clase entidad. Se utiliza la misma configuración, las relaciones tienen un ID y ese ID con qué ID de las entidades está relacionado.

6.8.3. Interfaz

Figura 8. Paquete Interfaz

El paquete Interfaz está compuesto por:

Mi JCanvas

Otra de las clases más importantes ya que en ella está programada las funciones, movimientos, y click´s del ratón.

- ElementoBajoMouse → Es el método más importante de la clase, con él se podrá saber si el mouse está situado encima de una entidad o de una relación, se jugará siempre con las posiciones (x,y). A partir de éste método se conseguirá realizar las diferentes opciones del mouse.

- MousePressed, MouseReleased, MouseEntered, MouseClicked… → Son los diferentes métodos con los que se va a poder pulsar una entidad y señalarla para poder arrastrarla/moverla de lugar en el panel, o desmarcarla para poder arrastrar a otra entidad y crear una relación. Se ha programado la función del clic derecho e izquierdo: con el izquierdo funciones de arrastrado, creación de entidades y pinchado de opciones y con el derecho obtener un menú pop-up cuando el mouse detecta estar encima de una entidad o una relación.

- CrearEntidad → Con este método se creará una entidad con dimensiones fijas en el panel una vez se pulse el botón y se pinche en el panel, se realiza una comprobación para que no se dibuje una entidad encima de otra, y se creará un ID no repetido para poder crear más tarde relaciones, búsquedas, o eliminar entidades o relaciones.

- CrearRelacion → Una vez creadas dos entidades, se localizará el punto inicial del arrastre con el método elementoBajoMouse y hasta donde se arrastra, guardado ambas posiciones y creando con esas posiciones la recta de lado a lado y de punto medio a punto medio. En la barra de status aparecerá la relación creada, el nombre y en qué sentido se dibujó.

- ActualizarMouse → Método para saber en todo momento donde está el mouse, y guardar su última posición, este método se utilizará constantemente.

- MarcarElemento → Se da un color a la entidad marcada una vez hecho click con el botón izquierdo sobre ella.

6.8.4. Recursos

En este paquete se guardan las imágenes utilizadas en la interfaz de la aplicación.

6.9. Manual de Usuario

Como se comentó anteriormente la aplicación está programada en lenguaje Java. Cuando se inicia la aplicación te da la posibilidad de crear un nuevo proyecto o cargar uno existente.

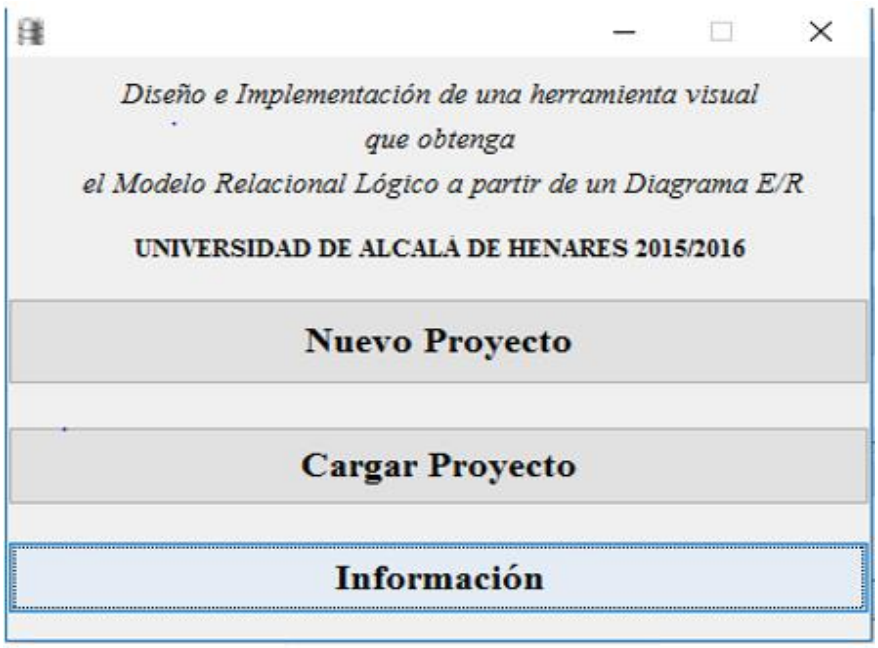

Figura 9. Panel de Inicio

Una vez cargado o creado un nuevo proyecto la pantalla de creación o modificación será la misma. Si se ha seleccionado cargar proyecto se abrirá a la vez una pantalla para seleccionar la ruta donde se ha guardado el fichero .dat.

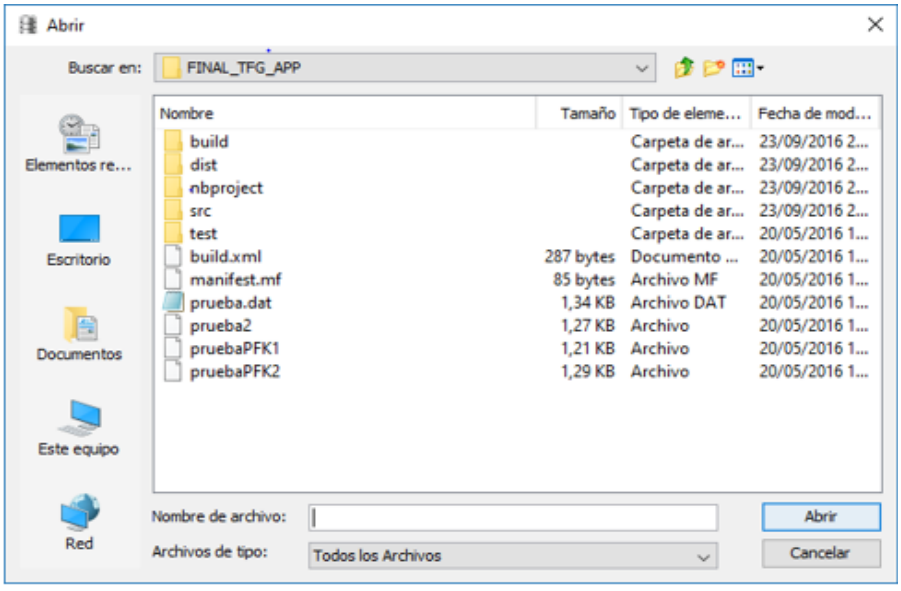

Figura 10. Selección de Fichero

Si se ha dado a nuevo proyecto se mostrará el panel de diseño directamente.

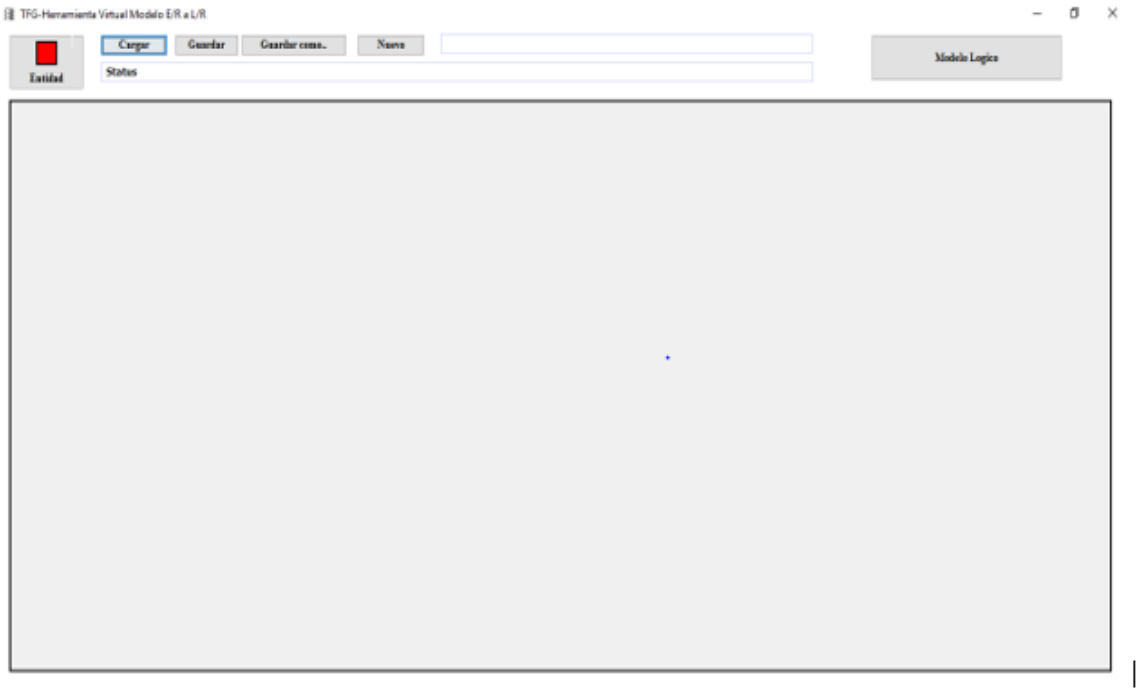

Figura 11. Panel de Diseño

Desde éste panel se tendrá opción también de iniciar un nuevo proyecto, dando la opción de guardar antes el actual si anteriormente se había cargado o comenzado a crear un nuevo proyecto.

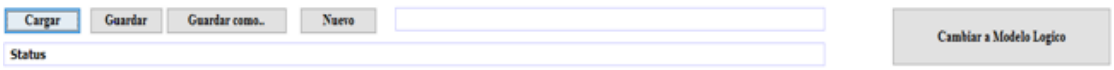

Figura 12 Opciones Panel

En la barra de “status” se irá informando, donde se ha guardado un proyecto, si es uno nuevo...

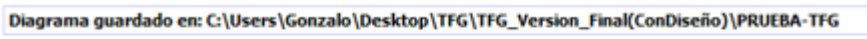

Figura 13. Barra de Status

A primera vista, el panel de diseño es simple y sencillo ya que la mayoría esta automatizado. El único botón será el de crear una entidad nueva, cada vez que el usuario lo seleccione se marque después en el panel, se creará una entidad con el mismo tamaño y forma establecida. El otro botón es el de conversión, cuando se marque o se desmarque se pasará de un modelo a otro.

A continuación, se pasa a explicar el uso y manejo de las entidades.

- Entidades

Figura 14. Panel de Diseño con Entidad

Cuando esté pinchado, se marcará en el panel y se creará una nueva entidad.

- Acciones:
  - Si se marca esta entidad y se pincha sobre ella y sin soltar se podrá moverla por el panel.

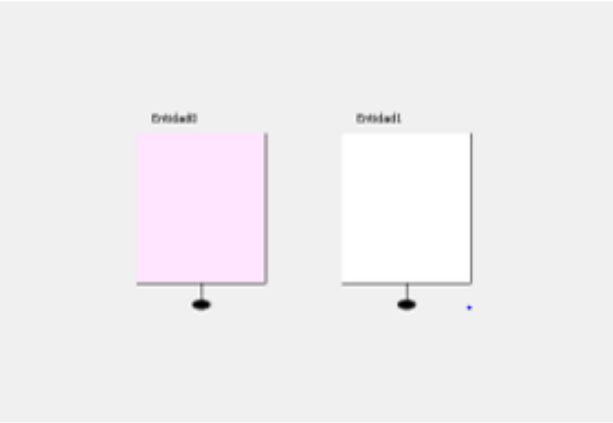

Figura 15. Entidades

- Si se pincha con el botón derecho saldrá un pop-up donde dará la opción de eliminar o entrar a la ventana de propiedades.

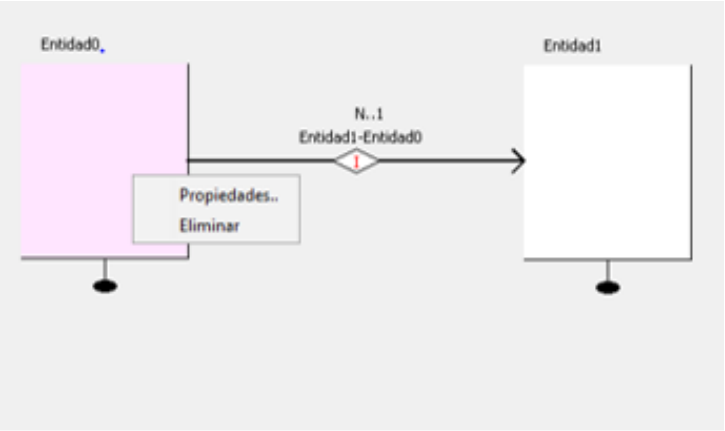

Figura 16. Menú Entidades

- Y por último si se ponen dos entidades, y sin marcar la entidad se pincha sobre una de ellas y se arrastra hasta la otra entidad, se creará la relación. Es importante la dirección del arrastre.

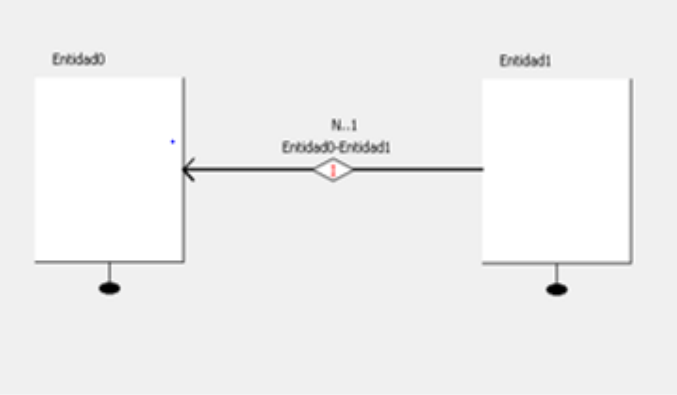

Figura 17. Crear Relación

La diferencia de mover la entidad o crear la relación, para que el programa entienda qué es lo que se quiere hacer a la hora de arrastrar el ratón, es la acción de marcar la entidad y cambiará de color para ver la diferencia.

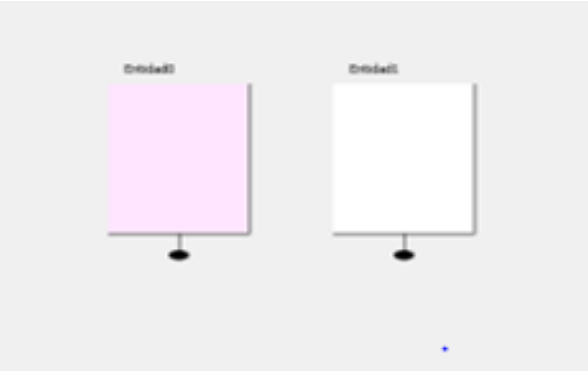

Figura 18 Diferencia Entidades

Si se elimina una entidad que tiene una relación con una o varias entidades, también se eliminaran las relaciones unidas a ella. En ningún caso se podrá ver una relación sin una de las entidades necesarias.

- Relaciones

Cuando se tienen dos o más entidades en el panel, se podrá crear relaciones (explicado en el punto anterior). Estas relaciones se fijarán en un punto de la entidad, lo que permite que cuando se mueva la entidad, la relación se mueva también. La línea que identifica la relación se creará entre las dos entidades en sus lados más próximos, facilitando que no se crucen líneas.

Como se ve, dibuja automáticamente una línea, y un rombo (símbolo de relación).

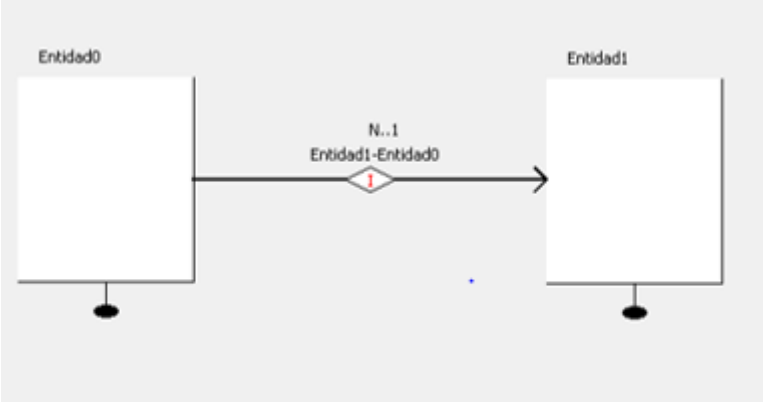

Figura 19 Relaciones

Según el tipo de relación se dibuja o el rombo vacio (No identificativa) o el rombo con una ”I“ (Identificativa).

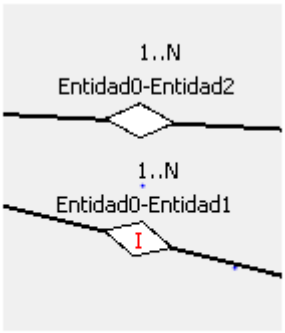

Figura 20. Tipos de Relación

- Acciones:

    - Si se pincha con el botón izquierdo encima del rombo, aparece un pop-up donde podrá eliminar o entrar a la ventana de propiedades de la relación.

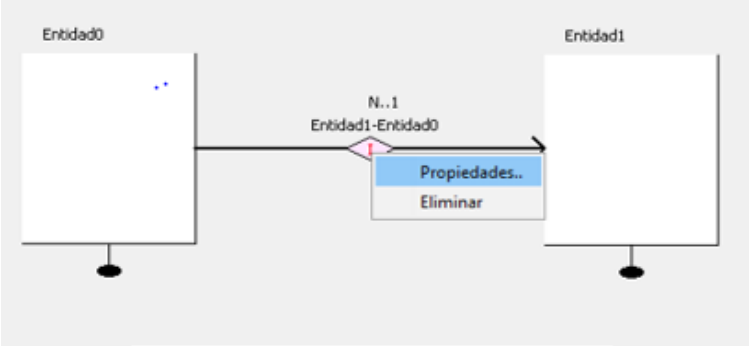

Figura 21. Menú Relaciones

    - Para mover la relación se tiene que mover la entidad.

Estará definido un tipo de relación y una cardinalidad por defecto. Es importante a la hora de hacer la relación saber si se quiere unir de la entidad - 1 a la entidad - 2 o de la entidad - 2 a la entidad - 1

Si se elimina una relación, solo se eliminará la relación sin tocar las entidades unidas a ella.

- Conversión

Éste botón lo únicamente se encargará de convertir lo que hay en el panel de diseño de un modelo a otro y viceversa. Se mantendrá en el modelo marcado hasta que el usuario lo desmarque, de esa manera volverá al anterior.

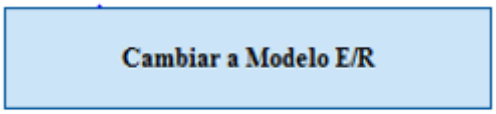

Figura 22. Conversión

- Pantalla propiedades Entidad.

Ésta será la pantalla encargada de transferir y dibujar todo lo escrito en ella en la entidad del panel. Acciones:

- Se escribirá el nombre de la tabla, la fusión de este nombre con el nombre de la entidad relacionada, será el nombre de la relación.

- Se insertarán los atributos y se debe señalar de qué tipo son. Cada vez que se inserta un atributo se añadirá en un panel para poder ir viéndolos. Para que un atributo sea PK de la entidad únicamente se tendrá que hacer doble-clic sobre él, y el atributo se añadirá al panel contiguo. También si se quiere deshacer éste cambio y que vuelva al panel de los atributos con hacer doble-clic automáticamente se moverá. También se puede señalar un atributo o PK introducida y eliminarlo pulsando el botón correspondiente.

- Al dar aceptar todos los datos se dibujarán en el panel, se podrá modificar cualquier dato sin necesidad de borrar toda la entidad.

Figura 23. Propiedades de la Tabla

- Pantalla propiedades Relación.

Esta pantalla podría ser un poco más compleja.

- Acciones:

- El nombre de la relación como se puede ver es la fusión de los nombres de las dos entidades relacionadas.

- Se deberá señalar el tipo de relación si es Identificativa o No Identificativa, esto se ve reflejado en el panel poniendo una I en el rombo o no.

- También se debe indicar el tipo de cardinalidad, si se señala la cardinalidad N…M se desbloquearán los paneles de los atributos, ya que con esta relación se crea una tabla auxiliar y podrá tener sus propios atributos. La PK será la fusión de las PK´S de las entidades unidas.

- Al igual que en la ventana de propiedades de la entidad, se añaden y se borran atributos igual.

- Al pulsar el botón de aceptar se dibujará automáticamente lo escrito en esta pantalla en el panel.

Figura 24. Propiedades de la Relación

Para finalizar el proyecto lo único que quedaría es guardar el archivo. Y observar los resultados obtenidos una vez construido el E/R y convertirlo al modelo Lógico Relacional.

- Ejemplo de Conversión

Como se ve a continuación se ha creado un proyecto – ejemplo de E/R a la vez que se ha ido explicando el funcionamiento de aplicación.

En nuestro caso se han creado tres entidades con relaciones N: M y N: 1 una de ellas identificativa y otra no identificativa para poder observar el paso de las claves primarias en función de las relaciones.

La clave primaria de Dirección es compuesta formada por tres atributos, y el resto de claves se formarían con un único atributo.

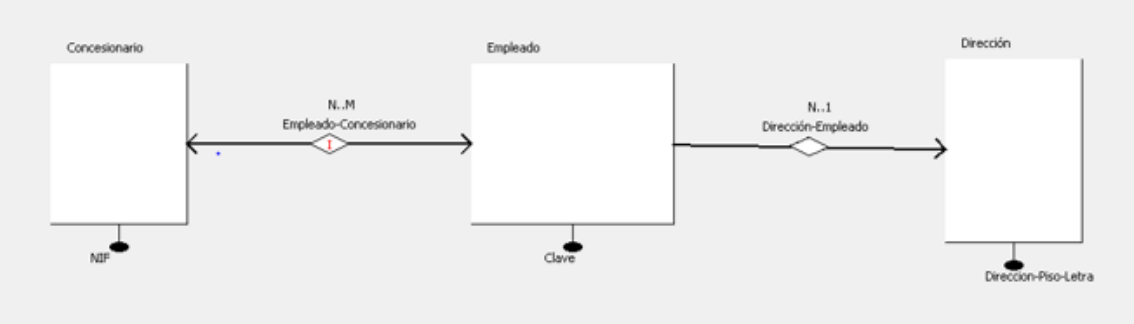

Figura 25. Ejemplo Conversión

Una vez convertido el diagrama E/R al modelo Lógico Relacional, se observa que en la relación N: M se ha creado una tabla intermedia, pasando las PK de las entidades relacionadas como PFK a la nueva tabla, y debido al tipo de relación en la 1:N la PK de la entidad empleado pasa a la entidad dirección como FK.

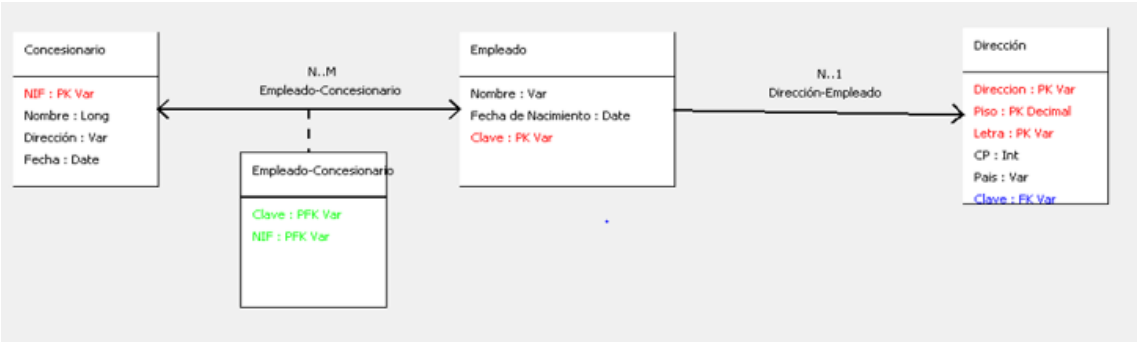

Figura 26. Ejemplo de Conversión 2

6.10. Conclusiones y trabajo futuro

Cuando acepté realizar este proyecto sabía que no iba a ser una tarea fácil, no consistía solo en investigar y aprender sobre los modelos y diagramas de base de datos, si no también aprender un nivel bastante elevado de lenguaje java el cual yo no tenía.

Tampoco dudé en elegir el proyecto de Iván no solo por la atención y dedicación que tiene con sus alumnos, sino porque a lo largo de la carrera las dos asignaturas de base de datos son las que más interesantes me han parecido y donde más he aprendido.

No era mi punto fuerte la programación, pero decidí optar por este lenguaje ya que para el sector laboral aprender más me podía venir bastante bien. Una vez finalizado éste proyecto puedo decir que mí nivel y conocimientos sobre este lenguaje han aumentado.

Al principio tenía bastantes dudas de cómo realizar partes de la aplicación, pero ya sea con la ayuda de mí tutor de cómo orientar los problemas, y de ejemplos en librerías he podido solventarlos. Uno de los problemas más significativos y que conseguí hacer funcionar fue la creación de las formas geométricas, conseguir que hacer que una línea recta se cortara justo en el perímetro de un rectángulo, o que se creara un rombo en el punto medio de la recta. Al final con fórmulas matemáticas y obteniendo la posición (x,y) continuamente se solucionó y conseguí el resultado querido. Este método para mí es el más significativo de la aplicación.

He llegado a cumplir mis objetivos, la aplicación funciona correctamente, pero como todo puede mejorarse o continuar en un futuro.

Se puede llegar a plantear la conversión a lenguaje SQL ya que una vez obtenido el modelo lógico relacional se podría generar el código SQL: únicamente sería ir leyendo los datos de las tablas y relaciones, el diseño de la aplicación la forma de acceder a los menús o que se visualizaran todos los atributos, podría ser una mejora, pero desde mi punto de vista preferí hacer un panel limpio y destacar las más importante. También se podría plantear introducir atributos multivaluados o añadir más simbología (entidad fuerte, entidad débil…).

Estas mejoras serían bastante interesantes para conseguir una aplicación completa y puede que única en el mercado, ni libre ni de pago.

Como se ha dicho anteriormente sí hay aplicaciones para realizar el diagrama E/R o el Lógico Relacional, pero por separado.

Se puede utilizar actualmente para el aprendizaje para la asignatura de Base de Datos de la Universidad de Alcalá de Henares, gracias a su conversión al instante de E/R a Lógico

Relacional y viceversa hace que sea más visible y poder comprender mejor las normas de conversión.

Y el propio código se puede reutilizar en otras aplicaciones parecidas, sobre todo la clase que pertenece al paquete Auxiliares, ya que es el pilar a la hora de construir las formas geométricas.

7. Anexo

8. Bibliografía

- **Apuntes teoría de las asignaturas de Base de Datos y Base de Datos avanzadas.**
- **https://docs.oracle.com/javase/7/docs/api/**
- **https://docs.oracle.com/javase/7/docs/api/java/awt/Canvas**